

メディアスケーリング可能な連続メディアオブジェクト

Continuous Media Objects with Media Scaling

中島達夫 *

Summary. Continuous media applications will be popular in future computing systems. For supporting various platforms, processing continuous media data should be scalable according to processing power of CPU, bandwidth of networks, and system loads. In this paper, we propose scalable continuous media objects as a framework for implementing felexible continuous media applications.

1 はじめに

最近のワークステーションは、ビデオキャプチャ、マイクロフォン、スピーカなどを備え付けているため、オーディオやビデオなどの連続メディアオブジェクトを扱うことが可能となってきた。また、PowerPCやAlpha等の高速CPU、大容量メモリ、安価な大容量ディスク、ATM等の高速ネットワークなどの従来のハードウェアの性能の向上、JPEGやMPEG等のデータ圧縮技術などの進歩にともない、高品質の連続メディアオブジェクトを扱うことも可能となりつつある。

我々のグループでは、連続メディアオブジェクトを扱うためのオペレーティングシステムである Real-Time Mach と、その上のストレージシステム、ネットワークシステムなどを開発してきた [6]。これまでの最も重要な課題は、実時間リソース管理技術を用いて、連続メディアオブジェクトの時間制約を保証することであった。しかし、様々な性能の計算機を用いたり、様々な負荷の状況でアプリケーションを実行することを考えると、マシンの性能が高く負荷が軽い状態では、十分連続メディアデータの時間制約を保証して実行することが可能であるが、負荷が重くなったり、マシンの性能が十分でない時は、時間制約を崩壊せずに連続メディアオブジェクトを処理することが困難であった。

我々は、このような問題点を解決する手段として、動的 QOS 制御を用いた連続メディアシステムを構築してきた [5, 8]。このシステムでは、メディアの品質を変更するメディアスケーリングをアプリケーションですべて行って

* Tatsuo Nakajima. 北陸先端科学技術大学院大学

いた。つまり、連続メディアオブジェクトをどのように処理するかはすべてアプリケーション/システムに任されていた。しかし、多くの場合、どのようにメディアスケーリングを行うかは、各連続メディアオブジェクトの性質に依存しているため、アプリケーション/システムで制御することが困難であった。

本稿では、メディアの品質をどのように変更可能であるかを、アプリケーションが決定するのではなく、連続メディアオブジェクトが持つべき性質であると考え、そのため、従来の連続メディアオブジェクトの表現法 [2, 4] を基に、適応可能な連続メディア処理を実現するためのフレームワークとして、メディアスケーリング可能な連続メディアオブジェクトを提案し、その実現法に関して議論する。

以下、第2節では、連続メディアオブジェクトの品質をユーザが指定するパラメータであるサービスの質とメディアスケーリング、適応的な連続メディア処理に関して述べる。第3節では、メディアスケーリング可能な連続メディアオブジェクトを定義する。第4節では、メディアスケーリング可能な連続メディアオブジェクトの実現法に関して述べる。

2 サービスの質とメディアスケーリング

本節では、連続メディアオブジェクトの直観的な説明と、アプリケーションがオペレーティングシステムに必要なリソース量を伝えるために用いるサービスの質 (QOS, Quality of Service) に関して述べる。次に、連続メディアオブジェクトの品質を変更可能にするメディアスケーリングに関して述べる。最後に、メディアスケーリング可能な連続メディアオブジェクトと適応的な連続メディア処理の関係に関して述べる。

2.1 連続メディアオブジェクトとサービスの質

連続メディアオブジェクト (Continuous Media Objects) は離散的なデータのストリームとして表現され、ストリームの構成要素である個々のデータオブジェクトに時間的制約が付随しているものとして定義する。個々のデータオブジェクトに与えられた時間的制約が同一の場合、そのストリームを周期的データストリームと呼び、そうでない場合を非周期的データストリームと呼ぶ。また、データストリームが、連続的に定義されていない場合を非連続データストリームと呼ぶ。例えば、ビデオ、動画、音声データなどは、周期的データストリームで表わせ、スライドショーのようなオンラインのデスクトップ・プレゼンテーションは、非周期的データストリームで表すことができる。また、手動によるスライドなどのプレゼンテーションは、非連続データストリームとなる。

デジタルビデオやオーディオのように連続的に変化する連続メディアオブジェクトは、データの正当性がその時の値の正当性 (Logical Correctness) だけではなく、いつその値に変化したかという時間的な正当性 (Temporal Correctness) にも依存している。このようなデータは、周期的なデータストリームにより表すことができ、厳密な時間制約を持つ連続メディア・アプリケー

ションでは、このようなデータを大きな遅れや損失なく処理する必要がある。連続メディア・データを正しく扱うため、オペレーティング・システムは、ユーザがどのような資源を必要としているかを実行される前に知る必要がある。ユーザの要求をオペレーティング・システムに伝える方法の1つが、サービスの質 (QOS, Quality of Service) をアプリケーションが記述することである。

連続メディア・データの扱いにおいては、サービスの質は、空間的解像度 (Spatial Resolution)、時間的解像度 (Temporal Resolution) により表現することができる。例えば、ビデオデータの場合、空間的解像度はデータサイズやデータ圧縮率により決定され、時間解像度はフレーム数/秒 (fps) により決定される¹。

各アプリケーションが指定した QOS を保証するためには、実時間処理技術が必要となる。つまり、各アプリケーションは指定された QOS を保証するために必要なリソースを実時間スケジューリングに基づいて割り当てる。この割り当てられたリソースは他のアプリケーションにより使用されることがないことが保証される必要がある。

2.2 メディアスケールリング

従来の連続メディアオブジェクトの表現では、メディアの品質は連続メディアオブジェクトが静的にもつ属性であり、動的に変化することができないものと考えられていた。この場合、オペレーティングシステムに渡される QOS は、処理する連続メディアオブジェクトにより決定され、ユーザが指定できるものではなかった。そのため、メディアの品質を変更したい場合には、複数の連続メディアオブジェクトを用意する必要があった。しかし、多くのビデオやオーディオなどのデジタル表現を考えると必ずしも連続メディアオブジェクトの品質は静的なものではないと考えることができる。

このような品質を動的に変更することが可能な連続メディアオブジェクトを、メディアスケールリング (Media Scaling) 可能な連続メディアオブジェクトと呼ぶ。連続メディアオブジェクトの品質を変化させるためのパラメータとして、いくつかの手段を考えることができる。例えば、ビデオのフレームレートや圧縮率、ピクセル毎のビット数、オーディオのサンプルあたりのビット数、ステレオ/モノラルの変換などである [1]。

簡単な例としては、ビデオストリームの適当なフレームを間引く (例えば、2 フレームのうちの 1 フレーム) ことによりメディアの品質を変化できる。また、各フレームを 4 ビット毎のブロックに分割し、そのうちの左上の 1 ビットと残りの 3 ビットを分割することにより、すべてのブロックの 1 ビットだけ用いれば品質の低いフレームを構成できる。あるいは、フレームの白黒部分と、カラー部分を分けたり、オーディオの左右を合成した音と、左右の差分の音に分割することにより同一の連続メディアオブジェクトの品質を様々に変化することが可能となる。

特に、最近のビデオの圧縮法などでは、サブバンドコーディングを用いたものや JPEG のプログレッシブモードやハイアラキカルモードや MPEG2 の

¹ ロスレートは時間解像度としてモデル化することが可能である。

スケーリング機能などを用いることにより容易にメディアスケーリングを実現することが可能となる。

2.3 適応的な連続メディア処理

実際のアプリケーションでは、各メディアのサービスの質は固定した1つのQOSを用いて指定するのではなく、動的に変更可能なQOSとして表現するとシステムの柔軟性が格段に向上する。システムが混雑している時は、重要度の低いアプリケーションが処理する連続メディアオブジェクトの品質を自動的に低下する。一方、システムの負荷が軽くなった時に連続メディアオブジェクトの品質を向上する。また、新しくアプリケーションを実行する時、アプリケーションが必要とする資源が現在使用可能な資源より多いときは、重要度が低くメディアの品質を下げる事が可能なアプリケーションを選択し、そのアプリケーションが処理する連続メディアオブジェクトの品質を下げる事により資源使用量を減少させ、新しいアプリケーションの実行を可能とする。

このように、メディアスケーリング可能な連続メディアオブジェクトを用いてサービスの質を動的に変化させる手法を動的QOS制御と呼ぶ。メディアスケーリング可能な連続メディアオブジェクトを用いて動的QOS制御を行なうことにより²、適応的な連続メディア処理が可能となる。メディアスケーリング可能な場合は、ユーザは実際に好ましいメディアの品質を決定する必要がある。なぜなら、ユーザは、連続メディアオブジェクトが提供可能な最低の品質に満足することができないかも知れないし、最高の品質を必要としないかも知れない。このように、ユーザが各自の好みをQOSのレンジとして指定することが可能となるので、連続メディアオブジェクトの品質とユーザの好みを分けることができるため、アプリケーションの柔軟性が格段に向上する。

マルチメディアマニュアルの再生を考えてみると、高品質のオーディオデータを再生するだけのCPU能力をワークステーションが持っていない場合は、オーディオデータのビット数を減らしたり、モノラルにしたり、オーディオデータを再生せず、そのオーディオデータと同じ内容のテキストデータを画面に表示することも可能である。ビデオデータに関しても同様にシステムの能力や負荷に応じて容易に連続メディアオブジェクトの品質を変更することができる。また、複合オブジェクトとして連続メディアオブジェクトを考えると、各連続メディアオブジェクトのそれぞれの品質を変化させるのではなく、あまり重要でない情報の連続メディアオブジェクトをユーザからの指定があるまで表示せず、ユーザからの要求があった時に、別な連続メディアオブジェクトの表示を止めて、指定された連続メディアオブジェクトを表示することにより、プレゼンテーション全体としての品質を動的に変化することも可能であると考えられる。

² 遅延が大きくなってもかまわない場合は、いくつかのバケットをまとめることにより、全体の処理を減らすことでスループットを変化させるようなメディアスケーリングを伴わない動的QOS制御を考えることもできるが、本稿では考慮しない。

3 メディアスケーリング可能な連続メディアオブジェクト

本節では、連続メディアオブジェクトとQOSを定義した後、メディアスケーリング可能な連続メディアオブジェクトを定義する。更に、複数のメディアスケーリング可能な連続メディアオブジェクトをコンポジットした連続メディアオブジェクトに関して議論する。

3.1 連続メディアオブジェクトの定義

連続メディアオブジェクトはデータオブジェクトが時間軸に沿って離散的に並べられたデータのストリームであると考えることができる。時間 t_1 にはじまり t_n に終る連続メディアオブジェクトを以下のように定義する。

$$CMO[t_1, t_2] = \langle i = t_1, t_2 | (D_i, M_i) \rangle$$

特に、時間0にはじまり、 n で終了する場合を $CMO[n] = \langle i = 0, n | (D_i, M_i) \rangle$ で表す。ここで、 M_i はデータオブジェクトであり、 D_i は、メディアデータ M_i が有効である時間とする。つまり、 M_i は、 $\sum_{j=t_1}^{i-1} D_j$ から D_i の間だけ有効なデータとなる。

この表記法を用いると、周期的データストリームは、 $\forall i, D_i = C \vee T_{i-1,1} = T_{i,0}$ として表現される。また、非周期的データストリームは、 $\forall i, T_{i-1,1} = T_{i,0}$ として表される。ここで、 $T_{i,0}$ を M_i が有効となる時間、 $T_{i,1}$ を M_i が無効になる時間であり、 $T_{i,0} = \sum_{j=t_1}^{i-1} D_j$ 、 $T_{i,1} = T_{i,0} + D_i$ となる。

最後に、連続メディアオブジェクトとアプリケーションがオペレーティングシステムにリソースの使用量を伝えるために用いるQOSとの関係について述べる。本節では、連続メディアオブジェクトはメディアスケーリング可能でないと考えている。そのため、連続メディアオブジェクトの品質を変更することはできないので、QOSは連続メディアオブジェクトの持つ品質と同一である必要がある。つまり、関数 $Q(CMO[n])$ がメディアの品質を表す関数であるとする、 $QOS = Q(CMO[n])$ となる必要がある。

3.2 メディアスケーリング可能な連続メディアオブジェクト

メディアスケーリング可能な連続メディアオブジェクトを定義する方法として、2つの方法を考えることができる。1つめの方法では、ストリームの各データオブジェクトである M_i の品質をインクリメンタルに変更する。2つめの方法では、ストリームに含まれる適当なデータオブジェクトを除いたサブセットのストリームを作る。その際、省かれたデータオブジェクトより早い有効時間を持ち、かつ、最後に処理されたデータオブジェクトの有効時間は、省かれたデータオブジェクトの有効時間を加えた時間有効の修正される。

1つめの方法を用いた場合は、 $CMO[t_1, t_2] = \langle i = t_1, t_2 | (D_i, M_i) \rangle$ で表される連続メディアオブジェクトの各データオブジェクト M_i を、 $M_i = \{M_{i,min}, M_{i,inc_1}, M_{i,inc_2} \dots M_{i,inc_{max}}\}$ で表す。ここで、 $M_{i,min}$ を最低の品質を保証するデータオブジェクト、 M_{i,inc_1} を $M_{i,min}$ の品質を1段階改善するために必要なデータとする。つまり、 $\{M_{i,min}, M_{i,inc_1}\}$ は $M_{i,min}$ を1段階改善したデータオブジェクト、 $\{M_{i,min}, \dots, M_{i,inc_{max}}\}$ は、 M_i で表現することができる最も品質の高いデータオブジェクトとなる。

この場合、最低の品質を表現する連続メディアオブジェクト CMO_{min} は以下のようになる。

$$CMO_{min}[t_1, t_2] = \langle i = t_1, t_2 | (D_i, M_{i,min}) \rangle$$

また、 CMO_{min} の品質を1段階だけ改善した $CMO_{min+1}[t_1, t_2]$ は、

$$CMO_{min+1}[t_1, t_2] = \langle i = t_1, t_2 | (D_i, \{M_{i,min}, M_{i,inc_1}\}) \rangle$$

と表すことができる。更に、連続メディアオブジェクトの最高の品質を表す $CMO_{max}[t_1, t_2]$ は、

$$CMO_{max}[t_1, t_2] = \langle i = t_1, t_2 | (D_i, \{M_{i,min}, \dots, M_{i,inc_n}\}) \rangle$$

と表すことができる。

以上から、連続メディアオブジェクト $CMO[t_1, t_2]$ は、

$$CMO[t_1, t_2] = \{CMO_{min}[t_1, t_2] \dots CMO_{max}[t_1, t_2]\}$$

と定義できる。これは、 $Q(CMO_{min}[t_1, t_2]) < Q(CMO_{min+1}[t_1, t_2]) < \dots < Q(CMO_{max}[t_1, t_2])$ となり、システムの性能や負荷に応じて適当な連続メディアオブジェクトの品質を選択することが可能であることを示す。

2つめの方法では、ストリーム中のデータオブジェクトの一部を削除する。ここで、 M_i を削除した場合を考える。この時、 M_{i-1} の有効期間は M_i の有効期間を加えたものとなる必要がある。そのため、 $D_{i-1}^{new} = D_{i-1}^{old} + D_i$ となる。ここで、いくつかのデータオブジェクトを削除した連続メディアオブジェクトを $CMO_{sub_1}[t_1, t_2]$ とし、これから更にいくつかのメディアオブジェクトを削除した連続メディアオブジェクトを $CMO_{sub_2}[t_1, t_2]$ とする。このとき、 $Q(CMO_{sub_2}[t_1, t_2]) < Q(CMO_{sub_1}[t_1, t_2]) < Q(CMO[t_1, t_2])$ となるなら、 $CMO[t_1, t_2]$ はメディアスケーリング可能である。しかし、品質をインクリメンタルに改善するようにサブセットを決定することは一般的に困難なことである。

最後に、ユーザにより指定された QOS とメディアスケーリング可能な連続メディアオブジェクトとの関係を考える。ここでは、ユーザが指定した QOS の最小値を qos_{min} 、最大値を qos_{max} とする。また、最低の品質のメディアオブジェクトを CMO_{min} 、最高の品質を CMO_{max} で表すと、連続メディアオブジェクトの最小の QOS を $qos_{CMO_{min}} = Q(CMO_{min})$ 、最高の QOS を $qos_{CMO_{max}} = Q(CMO_{max})$ で表す。

この時、実際に使用される最低の QOS は、 $qos_{min_{act}} = \max(qos_{min}, qos_{CMO_{min}})$ となり、最高の QOS は、 $qos_{max_{act}} = \min(qos_{max}, qos_{CMO_{max}})$ となる。

3.3 連続メディアオブジェクトのコンポジション

本節では、複数の連続メディアオブジェクトをコンポジットした複合連続メディアオブジェクトにおけるメディアスケーリングに関して議論する。複合連続メディアオブジェクトは、メディアを処理するためのツールキットやマルチメディアオーサリングを考える時、重要なものとなる。はじめに、接続、選択、並行、条件の4つのコンポジションをおこなうためのオペレータを定義する。

結合オペレータ \circ は、 $CMO_1[t_1, t_2] \circ CMO_2[t_3, t_4]$ のように使われ、 $CMO_1[t_1, t_2]$ の後に $CMO_2[t_3, t_4]$ を連結した連続メディアオブジェクトを意味する。つまり、 $CMO_{1 \circ 2}[t_1, t_2 + t_4 - t_3] = \langle i = t_1, t_2 | (D_i, M_i) \rangle \vee \langle i = t_3, t_4 | (D_i, M_i) \rangle$ となる。

選択オペレータ $|$ は、 $CMO_1[t_1, t_2] | CMO_2[t_3, t_4]$ のように使用され、 $CMO_1[t_1, t_2]$ と $CMO_2[t_3, t_4]$ のどちらか一方が処理されることを意味する。このとき、 $t_2 - t_1 = t_4 - t_3$ である必要がある。例としては、2つの連続メディアオブジェクトが日本語と英語の音声ストリームを表している場合、ユーザの指定によりどちらか一方を処理することが可能であると考えることができる。

並行オペレータ \parallel は、 $CMO_1[t_1, t_2] \parallel CMO_2[t_3, t_4]$ のように使用され、 $CMO_1[t_1, t_2]$ と $CMO_2[t_3, t_4]$ が同時に処理されることを意味する。この場合も、 $t_2 - t_1 = t_4 - t_3$ である必要がある。並行オペレータは、2つのビデオを同時に再生する場合や、オーディオとビデオを同時に再生する場合に用いられる。

条件オペレータは、 $test?CMO_1[t_1, t_2] : CMO_2[t_3, t_4]$ のように用いられる。ここで、 t を時間として、 $test(t)$ が真の間は $CMO_1[t_1, t_2]$ を、 $test(t)$ が偽となると $CMO_2[t_3, t_4]$ となる。ここでは、時刻 t に $test(t)$ が偽となった時、 $CMO_2[t_3, t_4]$ は、ストリームの最初から処理される。また、 $test(t)?CMO_1[t_1, t_1 + t] : CMO_2[t_2 + t, t_3]$ となる時は、 $test(t)$ が偽となった時、 $CMO_2[t_2, t_3]$ は、時間 t 経過した時点からスタートされる。条件オペレータはハイパーメディアを構築する際に有効であると考えられる。

次に、プレゼンテーションとフィルタ関数の2つの関数を考える。プレゼンテーション関数 $P(CMO[n], C)$ は、連続メディアオブジェクトを表示するための関数である。例えば、 $P((CMO_1[n_1] \parallel CMO_2[n_2]) | (CMO_3[n_3] \parallel CMO_4[n_4]), C)$ という関数は、 $CMO_1[n_1]$ と $CMO_2[n_2]$ を同時に表示するか、 $CMO_3[n_3]$ と $CMO_4[n_4]$ を同時に表示するかのどちらかを意味する。この例は、2チャンネルあるテレビを表していると考えることができる。この関数で用いられている、 C はクロックを表現している。クロックは時間の進み方を示す。 R が実時間であるとする時、 $C = R$ なら、ストリームは実時間のプレーバックとなる。 $C = -R$ なら、逆向きの再生であると考えることができる。また、 $C = \alpha \times R$ は早送りを、 $C = \alpha \times -R$ は巻戻しを意味する。

フィルタ関数は、 $CMO_1 = F_i(CMO_2, C)$ のように入力した連続メディアオブジェクトを変換する関数である。フィルタ関数は、エッジ検出やモーション検出などのイメージ処理や圧縮／伸長、データ変換をおこなう時に用いられる。

最後に、コンポジションされた連続メディアオブジェクトのメディアスケールリングを考える。コンポジットされた連続メディアオブジェクトは以下の2つの観点からメディアスケールリングを行うことが可能である。1番目は、複合連続メディアオブジェクトを構成する各エレメントの連続メディアオブジェクトをスケールダウンする。このとき、複数の連続メディアオブジェクトが並行オペレータにより結合されている場合は、どの連続メディアオブジェクトをスケールダウンするに関して考える必要がある。2番目は、選択オペレータで指定された2つの連続メディアオブジェクトのうち、連続メディアを処理する計算量にあわせてどちらかを選ぶ。つまり、マシンの能力が不足して

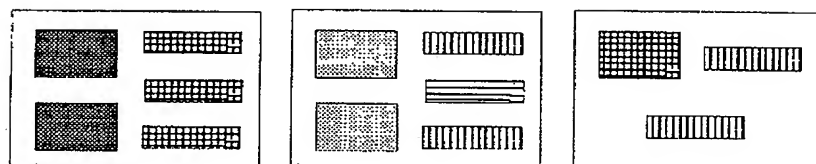


Figure 1. プレゼンテーションのスケーリング

いる場合は、より計算量を必要としない連続メディアオブジェクトの方を選択する。以上の方法により、コンポジットした連続メディアオブジェクトのメディアスケーリングを扱うことが可能となる（図1）。

3.4 適応的な連続メディアオブジェクトの表現

連続メディアオブジェクトを表現する方法として、オーディオデータの表現としては、ステレオの左右のデジタルデータを異なるストリームとして表現したり、1つのストリームにステレオの左右のデータを交互に配置する表現などが考えられる。また、ビデオデータに関しても、各フレームの各ピクセルのRGBを順々に並べて表現したものや（通常のRGB表現）、白黒部分とカラー部分を分けて（CCIR 601）順番に並べたものなどを考えることができる。しかし、2.2節で述べたメディアスケーリング可能な連続メディアオブジェクトを実現するためには、連続メディアオブジェクトの表現を工夫する必要がある。基本的な考え方としては、最低限の品質のメディアを表現するデータと、インクリメンタルにメディアの品質を向上するデータを分離することである。つまり、最低限の品質のメディアデータにインクリメンタルに品質を向上するデータを付け加えることによりメディアの品質を向上することが可能となる。

従来の連続メディアオブジェクトを表現する方法だと、ネットワークにおけるパケットの紛失や、バッファがあふれたとき、連続メディアアプリケーションでの処理が時間制約を満足することができないとき、ランダムにデータをドロップする可能性があるため、最低のQOSの保証が困難となる。我々の方式では、できるだけインクリメンタルにメディアの品質を向上するためのデータをドロップすることにより、最低限のメディアの品質を保証する。

以上の問題点を解決する、我々の提案する連続メディアオブジェクトの表現法では、各メディアの表現にレベルというメディアの重要度を導入することである。直観的には、最低の品質を保証するためのデータに最高のレベルを与え、インクリメンタルに品質を改善するためのデータのうち、より品質を改善するためのデータにより低いレベルを割り当てる。そこで、システムが、データをドロップする必要がある時は、よりレベルの低いデータから優先的にドロップする。図2では、一番上のメディアストリームが最も品質が高いメディアとなる。システムの負荷が思い時には、図に示すように、順番にレベルが低いデータをドロップしていくが、最低限の品質のメディアのみは表現できるようにする。

ここでは、レベルを表現する例として、以下の5つの場合を考える。1つ

Scale down of Media Data

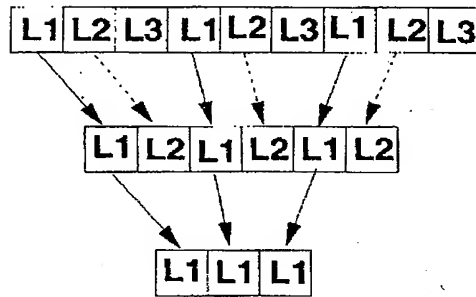


Figure 2. レベルの概念

めの例としては、30fps のビデオデータを考える。ここで、最低 15bps のデータは表示したいとするときは、2 フレーム毎のデータのレベルを残りのフレームより高くする。また、可能なら、20fps を再生したいとすると、残り 15 フレームのうち 5 フレームのレベルを他の 10 フレームより高くする。2 つめの例としては、JPEG のプログレッシブモードを用いた時、より精度の高いデータを再生するために必要なデータの部分のレベルを残りのデータの部分より低くする。3 つめの例としては、MPEG[3] を用いた時、I、P フレームを B フレームのレベルより高くすることが考えられる。また、4 つめの例としては、各フレームの白黒部分をカラー部分よりレベルを高くすることが考えられる。最後に、5 つめの例としては、ステレオオーディオの左右の和となるデータのレベルを左右の差分のデータより高くすることが考えられる。

以上のどの例の場合でも、より高いレベルのデータのみを処理することにより容易にメディアスケールリングを行なうことが可能となる。つまり、レベルの高いデータを処理することができない場合でも、レベルの低いデータの処理を行なうことにより、メディアの品質の低下をシステムの負荷に比例して順次落すことができるようになり、データのドロップによる予測不可能なメディアの品質の低下が発生しなくなる。

一般的にメディアの品質をどのように変化させるかに関しては、アプリケーションでは決めることができず、連続メディアオブジェクトの性質に依存したものであると考えられる。連続メディアの性質に関して詳しく知らないプログラマが、システムの負荷により、連続的にフレームレートを低下させるアプリケーションを作ったとする。連続的にフレームレートの変更により、品質の変更は連続的に行われるわけではないことが [9] で示されているように、連続的なフレームレートの変更では、正しくメディアスケールリングを行うことができない。

また、我々の提案する方式では、ビデオの性質に依存したメディアスケールリングのポリシーをアプリケーションの動的 QOS 制御のメカニズムと切り離すことができるため、制御の柔軟性が格段に向上する。例えば、動きが静的なビデオストリームでは、レベルが低いフレームを増やし、動きが激しいビ

デオでは、レベルが低いフレームを減らすことにより容易に連続メディアオブジェクトの意味を利用した柔軟なメディアスケーリングのポリシをアプリケーションを変更せずに採用することが可能となる。

4 結論

本稿では、適応的な連続メディア処理を可能とするメディアスケーリング可能な連続メディアオブジェクトとその実現法に関して述べた。メディアスケーリング可能な連続メディアオブジェクトを導入することにより、システムの性能や負荷に依存せず連続メディアオブジェクトを再生することが可能となる。

我々のグループでは、現在、Real-Time Mach 上に、連続メディアシステムのためのストレージシステム、ネットワークシステムを構築し、それらを用いて簡単なビデオオンデマンドシステムを作って動的 QOS 制御の実験を行っている。現在、この実験により、実時間リソース管理の重要性、動的 QOS 制御の有効性が実証されている [5, 8]。

また、本論文で述べたメディアスケーリング可能な連続メディアオブジェクトを実現するために必要な、リソース予約管理システム、ネットワークプロトコル、レベルの概念を提供するツールキットは現在実装中である [7]。

参考文献

- [1] L.Delgrossi, C.Halstrick, D.Hehmann, R.Herrtwich, O.Krone, J.Sandvoss, C.Vogt, "Media Scaling for Audiovisual Communication with the Heidelberg Transport System", In Proceedings of the ACM Multimedia'93, 1993.
- [2] A. Duda, R. Weiss, and D. K. Gifford, "Content-Based Access to Algebraic Video", Proceedings of the International Conference on Multimedia Computing and Systems, 1994.
- [3] D.Le.Gall, "MPEG: A Video Compression Standard for Multimedia Applications", CACM, ACM, Vol.34, No.4, 1991.
- [4] R.G.Herrtwich, "Time Capsules: An Abstraction for Access to Continuous-Media Data", IEEE 11th Real-Time System Symposium, 1990.
- [5] T.Nakajima and H.Tezuka, "A Continuous Media Application supporting Dynamic QOS Control on Real-Time Mach", ACM Multimedia'94, 1994.
- [6] 中島 達夫, "Real-Time Mach の現状と将来", JAIST Research Report, IS-TR-94-00S, 1994
- [7] 中島 達夫, "適応的な連続メディアオブジェクトを扱うためのオペレーティングシステム", JAIST Research Report, IS-TR-95-00S, 1995.
- [8] T.Nakajima, and H.Tezuka, "Experiences with a Continuous Media System supporting Dynamic QoS Control on a Microkernel-based Real-Time Operating System", Submitted for publication.
- [9] 手塚 宏史, 中島 達夫, "ビデオデータの滑らかな表示に関する考察", 第 11 回ソフトウェア科学会全国大会, 1994.